Internet 的安全套接层协议 SSL 协议剖析

张汛涞 (南京通信工程学院 210016)
张明杰 (南京军区司令部自动化站 210016)

摘要:本文主要分析 SSL 协议的内部结构和功能,并给出了主要的数据结构和参数,为利用 SSL 协议作网上系统开发和加密分析的用户提供一个参考。

关键词:套接层协议 信息安全 SSL

一、SSL 协议简介

SSL 协议的主要目标是在两个通信应用之间提供私有性和可靠性。协议由两层组成,最低层是 SSL 记录层协议(SSL Record Protocol),它基于可靠的传输层协议(如 TCP/IP),用于封装各种高层协议。高层协议主要包括 SSL 握手协议(SSL Handshake Protocol)、修改密码参数协议(Change Cipher Spec Protocol)、报警协议(Alert Protocol)、应用数据协议(Application Data Protocol)等。SSL 的优点之一是它独立于应用层协议,高层协议可基于 SSL 进行透明的传输。

SSL 协议提供的安全连接具有以下三个基本特性:

·连接具有私有性。在初始化连接后,制定密钥进行加密,对称密码体制用于数据加密(如:DES、RC4 等);

·对端实体鉴别可采用非对称密码体制或公开密钥密码体制(如:RSA、DSS);

·连接是可靠的。消息传输包括使用加密 MAC 算法进行消息完整性检查,MAC 计算中采用 HASH 功能(如:SHA、MD5)。

二、会话状态和连接状态

在介绍各协议之前,首先介绍会话和连接。

会话(session)是指客户和服务器之间的联系。会话由握手协议创建,定义了一套安全加密参数,为多个连接所共享。SSL 握手协议的职责是协调客户和服务器的状态,使得双方在不能精确地并行的情况下,也能工作一致。会话状态分为两种,一个是当前"操作态",另一个在握手协议期间,称之为"待决态"。另外,每种又分为读状态和写状态。当客户或服务器收到"change cipher spec"的信息,它把"待决读状态"拷贝到"当前读状态";当客户或服务器发送"change cipher spec"的信息,它把"待决写状态"拷贝到"当前写状态"。当握手协议结束后,客户和服务器交换"change cipher spec"消息,然后采用新达成的密码进行通信。

三、记录层协议

SSL 记录层以任意大小的非空块从高层接收尚未解释的数据。它主要有以下三个功能:

1. **分块**(Fragmentation)

记录层将信息分割成不超过 214 字节的 SSL 明文记录。客户信息的边界不在记录层中保存(即具有相同 ContentType 的多个客户信息可合并成一个 SSL 明文记录)。其结构如下:

```
struct {
    unit8 major, minor;
} ProtocolVersion;
enumb {
    change-cipher-spec(20), alert(21), handshake(22);
    application-data(23), (255)
} ContentType;
struct {
    ContentType type;
    ProtocolVersion version;
    unit16 length;
    opaque fragment(SSLPlaintext.length);
} SSLPlaintext;
```

2. **记录的压缩和解压**(Record Compression & Decompression)

所有记录采用在"当前会话状态"中定义的压缩算法进行压缩。压缩算法将 SSL 明文结构翻译成 SSL 压缩结构。压缩不能引起信息丢失,也不能使内容增加超过 1024 字节。如果解压功能使解压后长度超过 214 字节,则会产生错误 decompression-failure alert(压缩失败报警)。

其结构如下:

```
struct {
```

```
    ContentType type;
    ProtocolVersion version;
    unit16 length;
    opaque fragment(SSLCompressed.length);
} SSLCompressed;
```

3.记录有效负载保护和密码参数(Record Payload Protection & CipherSpec)

所有记录通过当前 CipherSpec 中定义的加密算法和 MAC 算法进行保护。一旦握手结束,两个群体共享为记录加密的密码,并计算信息鉴别码 MAC。加密和 MAC 操作将 SSL 压缩结构翻译成 SSLCiphertext。解密操作作相反的处理(传输时也包括序列号,这样可以发现信息是否丢失、修改或增加)。

其结构如下:

```
struct {
    ContentType type;
    ProtocolVersion version;
    unit16 length;
    select (CipherSpec.cipher-type) {
      case stream:GenericStreamCipher;
      case block: GenericBlockCipher;
    } fragment;
} SSLCiphertext;
```

四、修改密码属性协议

修改密码属性协议用于标识信号的转换。它只包含一条信息,信息内容为一个字节,其值为 1。其结构为:

```
struct {
    enumb {change-cipher-spec(1),(255)} type;
} ChangeCipherSpec;
```

"修改密码参数"信息由客户机和服务器双方发出,以通知接收方下面的记录将受到刚达成的密码参数(CipherSpec)和密钥的保护。当客户机或服务器收到 Change Cipher Spec 信息时,它将 pending read state(读待决状态)拷贝到 current read state(读当前状态);当客户机或服务器写 Change Cipher Spec 信息时,它将 pending write state 拷贝到 current write state。客户端在握手密钥交换和验证信息后发送 Change Cipher Spec 信息,服务器在成功处理了握手密钥交换信息后发送该信息。

五、报警协议

报警协议是 SSL 记录层协议的 content-type 所支持类型中的一种。报警信息包括警告内容和严重性级别。如果级别是 Fatal(致命错),则会导致当前的连接立即中断。在这种情况下,与该会话相连的其他连接可以继续进行,但其会话标识为无效,以防止它再建立新连接。

报警协议的结构(略)。

报警信息分为以下两种:

·Closure alerts(关闭报警)

客户机和服务器必须及时知道连接何时结束,以防信息被突然截断。报警协议中的 close-notify 消息通知接收方在本次连接中发方信息已发送完。

·Error alerts(错误报警)

在 SSL 握手协议中错误处理很简单。当发现一个错误后,发现方向其他群体发送一条信息。如果发送或收到的是致命错信息,则双方立刻关闭连接。服务器和客户端必须忘记有关这次连接的会话标识、密钥、密码等。在上述报警协议的结构中所定义的报警类型除 close-notify 外,其余 11 种均为错误报警。

六、握手协议

握手协议在 SSL 记录层之上,它产生会话状态的密码参数。当 SSL 客户端和服务器开始通信时,他们协商一个协议版本、选择密码算法、对彼此进行验证、使用公开密钥加密技术产生共享秘密。这些过程在握手协议中进行,其过程如图所示:

```
客户端                          服务器
ClientHello          - - - >
                                ServerHello
                                Certificate
                                ServerKeyExchange
                                CertificateRequest
                     < - - -    ServerHelloDone
Certificate
ClientKeyExchange
CertificateVerify
[ChangeCipherSpec]
Finished             - - - >    [ChangeCipherSpec]
                     < - - -    Finished
Application Data     < - - - >  Application Data
```

说明如下:

在客户端发送 client hello 信息后,对应的服务器回应 server hello 信息,否则产生一个致命错误,导致连接的失败。client hello 和 server hello 用于在客户端和服务器之间建立安全增强功能,并建立:协议版本号、会话标识符、密码组和压缩方法。此外,产生和交换两组随机值:

ClientHello.random 和 ServerHello.random。

在 hello 信息之后，如果需要被确认，服务器将发送其证明信息。此外，如果需要，可发送一个 server key exchange 信息。如果服务器被确认，并且适合于所选择的密码组，就需要对客户端请求证明信息。

现在，服务器将发送 server hello done 信息，表示握手阶段的 hello 信息部分已经完成，服务器将等待客户端响应。

如果服务器已发送了一个证明请求(certificate request)信息，客户端可回应证明信息或无证明(no certificate)告警。然后发送 client key exchange 信息，信息的内容取决于在 client hello 和 server hello 之间选定的公开密钥算法。如果客户端发送一个带有签名能力的证明，服务器发送一个数字签名的 certificate verify 信息用于检验这个证明。

这时，客户端发送一个 change ciher spec 信息，将 pending Cipher Spec(待决密码参数)拷贝到 current Cipher Spec(当前密码参数)。然后客户端立即在新的算法、密钥和密码下发送结束(finished)信息。对应地，如果服务器发送自己的 change ciher spec 信息，并将 pending Cipher Spec 拷贝到 current Cipher Spec，然后在新的算法、密钥和密码下发送结束(finished)信息。这一时刻，握手结束，客户端和服务器可开始交换其应用层数据。

Handshake message(握手信息)的结构(略)。

下面对 HandshakeType 的各类信息作一介绍。

·Hello Request(问候请求)

服务器可在任何时候发该信息，如果客户端正在进行握手协议，则该信息被忽略；如果客户端有空，则发送 Client Hello 信息。在随后的握手协商结束后，服务器才能再次发送 Hello Request 信息。

·Client Hello(客户端问候)

当客户端第一次连接到服务器时，应将 Client Hello 作为第一条信息发给服务器。Client Hello 包含了客户端支持的所有压缩算法，如果服务器均不支持，则本次会话失败。

·Server Hello(服务器问候)

Server Hello 信息的结构类似 Client Hello，它是服务器对客户端的 Client Hello 信息的回复。

·Server Certificate(服务器证明)

如果要求验证服务器，则服务器立刻在 Server Hello 信息后发送其证明 Certificate。Certificate 的类型必须适合密钥交换算法，通常为 X.509.v3 Certificate 或改进的 X.509 Certificate。

·ServerKeyExchange(服务器密钥交换)

当服务器没有 Certificate，或有一个仅用于签名的 Certificate(如 DSS，RSA)，或采用 Fortezza/DMS 卡，则需要使用 ServerKeyExchange 信息。

·CertificateRequest(证明请求)

如果和所选密码组相适应，服务器可以向客户端请求一个证明(Certificate)。如果服务器是佚名的，则在请求客户端 Certificate 时会导致致命错。

·ServerHelloDone(服务器问候结束)

服务器发出该信息表明 Server Hello 结束，然后等待客户端响应。客户端收到该信息后检查服务器提供的 Certificate 是否有效，以及服务器的 Hello 参数是否可接受。

·Client Certificate(客户端证明)

该信息是客户端收到服务器的 ServerHelloDone 后可以发送的第一条信息。只有当服务器请求 Certificate 时才需发此信息。如果客户端没有合适的 Certificate，则发送“没有证明”的告警信息，如果服务器要求有“客户端验证”，则收到告警后宣布握手失败。

·clientKeyExchange(客户端密钥交换)

信息的选择取决于采用何种公开密钥算法。参见上述 ServerKeyExchange。

·CertificateVerify(证明检查)

该信息用于提供客户端 Certificate 的验证。它仅在具有签名能力的客户端 Certificate 之后发送。

·Finished(结束)

该信息在 Change Cipher Spec 之后发送，以证明密钥交换和验证的过程已顺利进行。发方在发出 Finished 信息后可立即开始传送秘密数据，收方在收到 Finished 信息后必须检查其内容是否正确。

七、应用数据协议

应用数据信息由记录层处理，在当前连接状态下进行分块、压缩、加密，对于记录层来说信息是透明的。

八、结束语

以上介绍了 SSL 协议的主要组成、关系和数据结构，该协议由 Netscape 研制开发，目前最新版本为 V3.0。笔者提供两个网上地址，可以查阅 SSL 协议的详细资料：http://home.netscape/eng/ssl3/draft302.txt

http://home.netscape/eng/ssl3/index.html

(来稿时间：1998 年 9 月)